IE 498 - HW2

Julius Olson

Result

An accuracy of 97.9% was reached using the following settings.

Param	Value
Filter Size	`3 x 3`
Channels	`8`
Epochs	`10`
LR	`0.01`
Activation	`RELU`

Implementation

The class `CNN` is the main part of the program and contains methods for initialization, training and testing of the Convolutional network. The training is carried out via stochastic gradient descent. The forward and backward steps implemented follow the ones presented in the lecture slides available on Piazza.

Methods

- `train` - train the network
- `test` - test on supplied dataset
- `save` - Save the networks parameters to `.npy` file
- `load` - Load pre-trained network
- `init_params` - Initialize (Xavier init)

For the convolutions, an implementation of the vectorized version as presented by XX, was done (see `vec_conv` in the code). The difference from the algorithm presented in the paper is the order of the columns in the constructed matrices. This was done to facilitate for easier transformations, but makes no difference for the performance or result of the algortithm.

Instructions & requirements

```
# Train
python3 cnn.py train
# Load and test (default modelname = model.npy in current dir)
python3 cnn.py load

usage: cnn.py [-h] [--channels CHANNELS] [--filter FILTER_DIM]
              [--epochs EPOCHS] [--data DATA] [--model MODEL]
              {train,load}

Neural Network

positional arguments:
  {train,load}         Run NN training or load exisiting model

optional arguments:
```

```
-h, --help           show this help message and exit
--channels CHANNELS  Number of channels
--filter FILTER_DIM  Filter dimensions
--epochs EPOCHS      Number of epochs
--data DATA          Dataset path
--model MODEL        .npy model path
```

Requires python ver `3.6` or higher (due to usage of f-string formatting) and numpy.